ND-GAr: Custom Magnetic Field.

ND Sim/Reco Physics group meeting

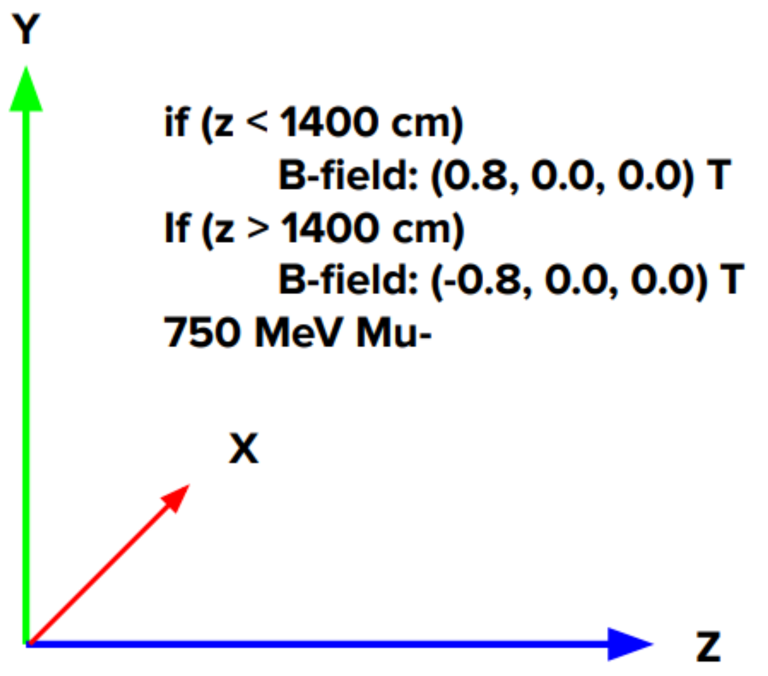
Eldwan Brianne for
ND-GAr group
10th February 2021

Custom Magnetic Field in edep-sim/GArSoft.

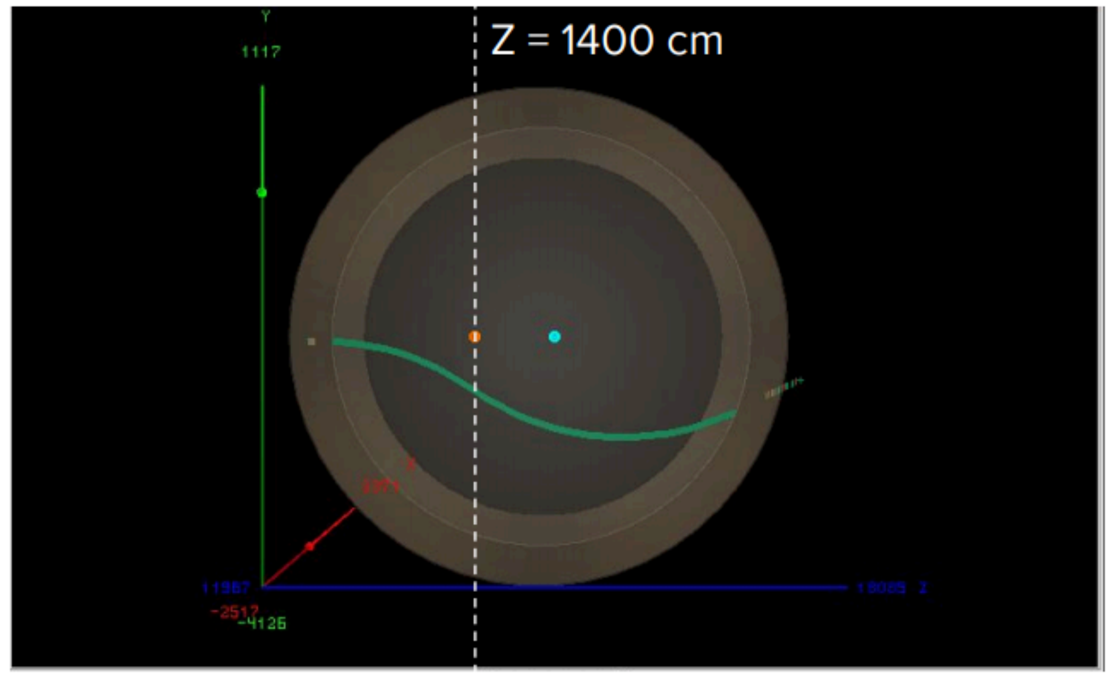
Implementation

- Due to the ND-GAr magnet, we need to use a custom magnetic field to simulate non-uniformities and see the impact on tracking performance (also may be good to understand the impact of the fringe field on the ND-LAr)
- Andrew Cudd nicely implemented this in
  - edep-sim (see PR: https://github.com/ClarkMcGrew/edep-sim/pull/10)
  - GArSoft as an art service (MPDMagneticField_service.cc)
- He demonstrated that it works perfectly with edep-sim
- GArSoft is currently in code-review (branch bfield)
- Current issue
  - Geant4 helper in GArSoft (to call G4 and pass the geometry/physics etc…)
  - Does not handle nicely new mag field service, no custom B-Field
  - -> Tracked to nug4 package

Event Displays

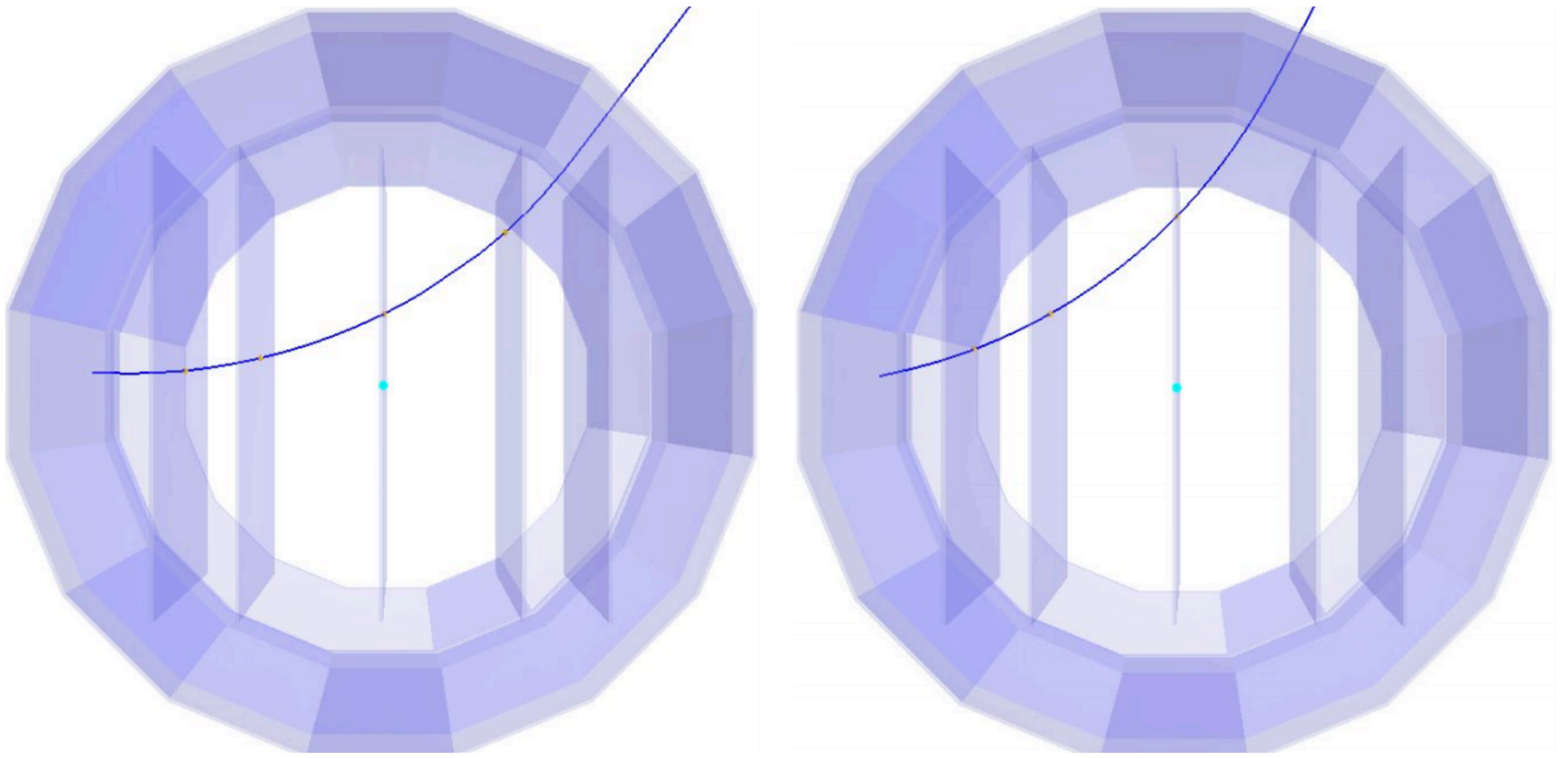


if (z < 1400 cm)
B-field: (0.8, 0.0, 0.0) T
If (z > 1400 cm)
B-field: (-0.8, 0.0, 0.0) T
750 MeV Mu-



Event Displays!

Simulated with SPY magnetic field map. Yes, these tracks now bend upward. The SPY field map is actually nominally -0.5 T in the X direction.

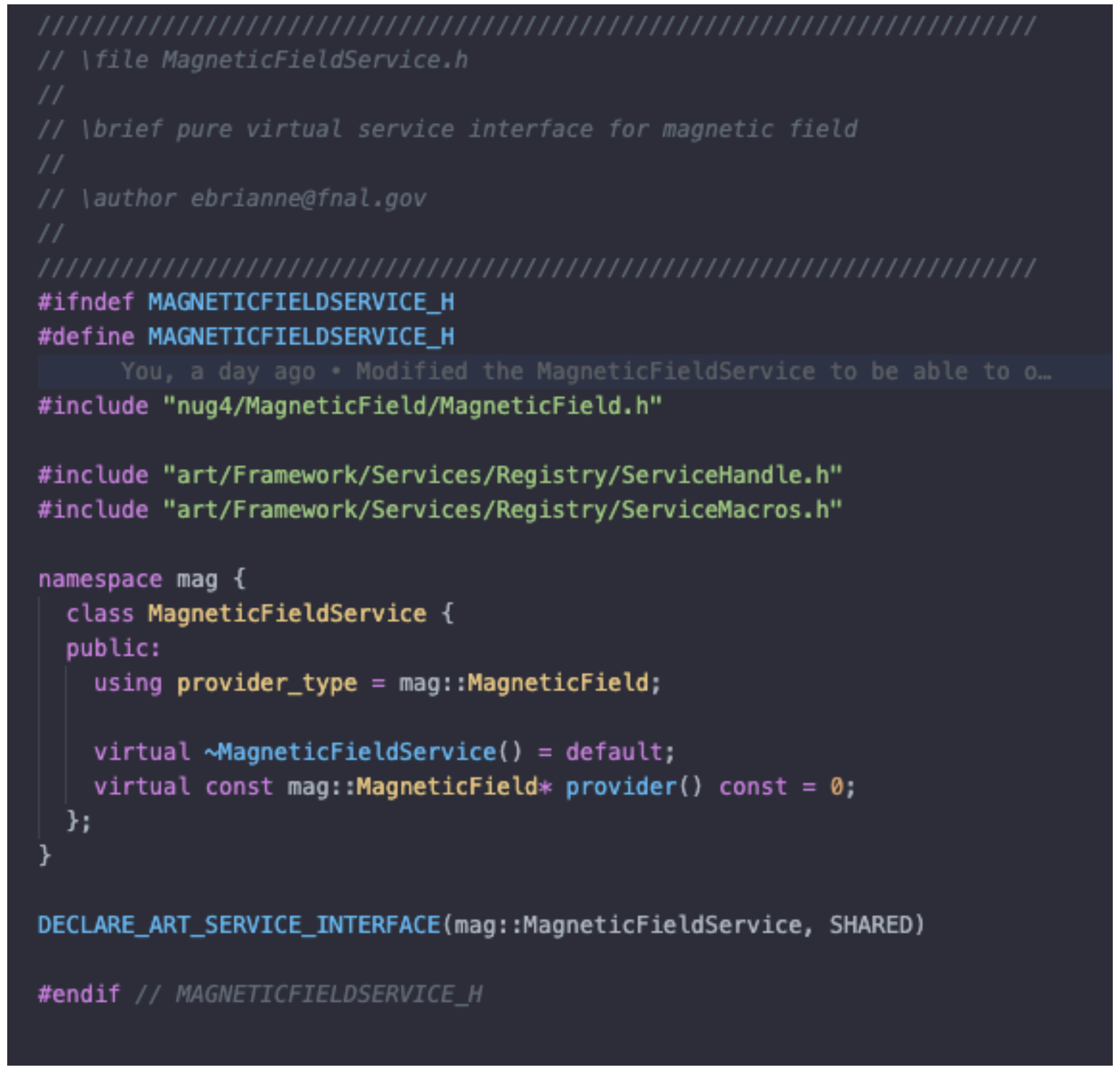


9

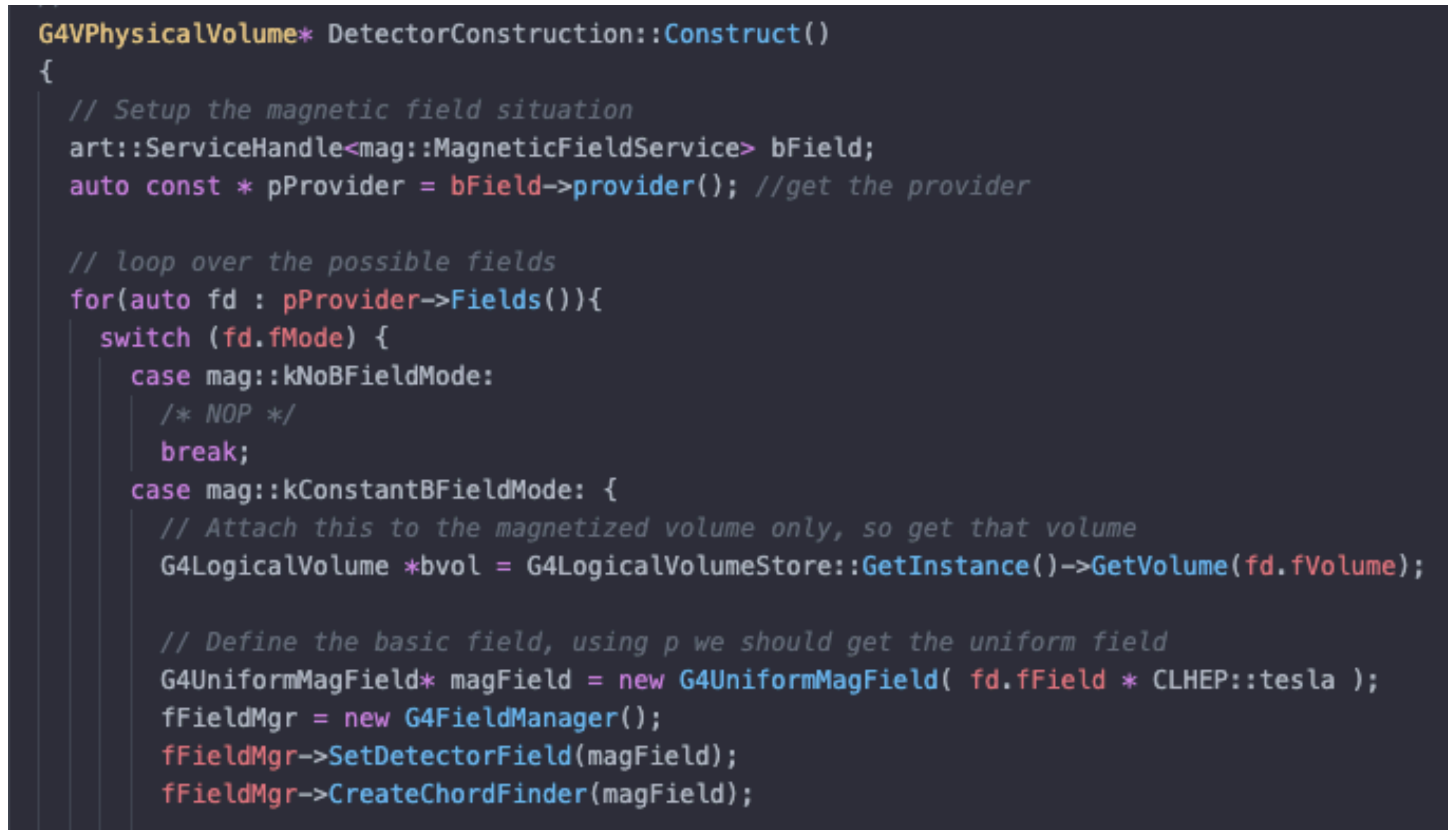
Modifications of nug4 package.

Changes

- The nug4 package handles a generic art Magnetic field service and the geant4 detector construction (read gdml and pass it to geant4)
- The Magnetic Field service currently cannot be overridden by custom service (via art service interface impl)
  - -> Implement a Magnetic Field service interface (abstract provider class)
  - -> Can be loaded by custom service using fhicl service_provider variable
- The Detector construction handles only uniform magnetic fields
  - -> Need to handle custom magnetic fields
- Modifications in review (see nug4 feature/MagneticField branch)
  - -> Implemented abstract service implementation and abstract MagneticField class
  - -> Implemented default magnetic field class and service to keep backward compatibility for NOvA and LArSoft
  - -> To implement custom magnetic field class for detector construction



```
////////////////////////////////////////////////////////////////////////
// \file MagneticFieldService.h
//
// \brief pure virtual service interface for magnetic field
//
// \author ebrianne@fnal.gov
//
////////////////////////////////////////////////////////////////////////
#ifndef MAGNETICFIELDSERVICE_H
#define MAGNETICFIELDSERVICE_H
      You, a day ago • Modified the MagneticFieldService to be able to o…
#include "nug4/MagneticField/MagneticField.h"

#include "art/Framework/Services/Registry/ServiceHandle.h"
#include "art/Framework/Services/Registry/ServiceMacros.h"

namespace mag {
  class MagneticFieldService {
  public:
    using provider_type = mag::MagneticField;

    virtual ~MagneticFieldService() = default;
    virtual const mag::MagneticField* provider() const = 0;
  };
}

DECLARE_ART_SERVICE_INTERFACE(mag::MagneticFieldService, SHARED)

#endif // MAGNETICFIELDSERVICE_H
```



```
G4VPhysicalVolume* DetectorConstruction::Construct()
{
  // Setup the magnetic field situation
  art::ServiceHandle<mag::MagneticFieldService> bField;
  auto const * pProvider = bField->provider(); //get the provider

  // loop over the possible fields
  for(auto fd : pProvider->Fields()){
    switch (fd.fMode) {
      case mag::kNoBFieldMode:
        /* NOP */
        break;
      case mag::kConstantBFieldMode: {
        // Attach this to the magnetized volume only, so get that volume
        G4LogicalVolume *bvol = G4LogicalVolumeStore::GetInstance()->GetVolume(fd.fVolume);

        // Define the basic field, using p we should get the uniform field
        G4UniformMagField* magField = new G4UniformMagField( fd.fField * CLHEP::tesla );
        fFieldMgr = new G4FieldManager();
        fFieldMgr->SetDetectorField(magField);
        fFieldMgr->CreateChordFinder(magField);
```

Backup Slides.